

## Extended Authorization Framework

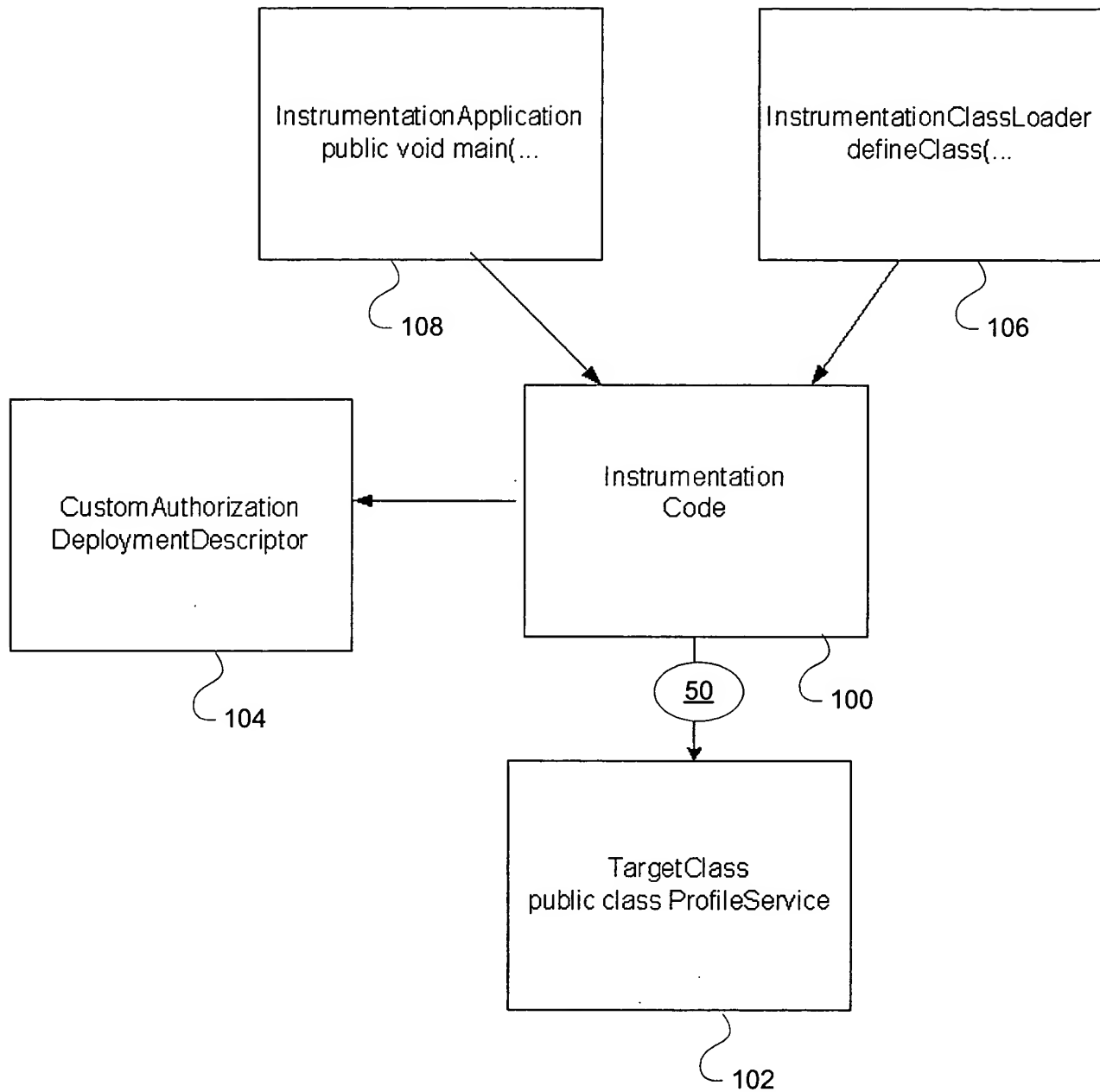


Fig. 1

### **Target Class ProfileService**

```
public Profile getProfile(String profileName, String attributes)
{
    System.out.println("\n*** ProfileService.getProfile entry ***");

    //get attributes associated w/profileName from DB
}
```

102a

**Fig. 2**

### Custom Authorization Deployment Descriptor

```
...
<authorizationTarget class="ProfileService">

    <subjectFactory class="ThreadSubjectFactory"/>

    <method name="getProfile">
        <requiredPermission class="com.ibm.resource.security.auth.ProfilePermission">
            <property name="name" value="UserProfile" />
            <property name="actions" value="Read" />
            <property name="attributes" value="attributes" scope="local" />
        </requiredPermission>

        <privilegedAction class="ReadAction">
            <property name="attributes" value="attributes" scope="local" />
        </privilegedAction>
    </method>

</authorizationTarget>
...
```

104a

Fig. 3

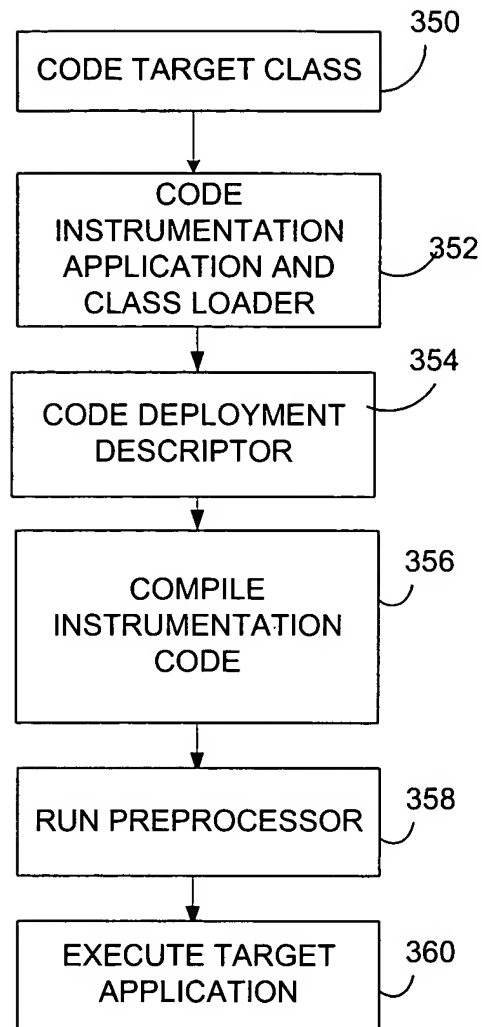


Fig. 4

## Target Class ProfileService — Instrumented Code

```
public Profile getProfile(String profileName, String attributes)
{
    System.out.println("\n*** ProfileService.getProfile entry ***");

    /* JAVA LANGUAGE EQUIVALENT OF INSTRUMENTATION
    (It's actually inserted into the target as byte code)
    */
    SubjectFactory subjectFactory= new ThreadSubjectFactory();
    Subject user=subjectFactory.getSubject();

    if(user != null && (System.getProperty("custom_authorization") != null))
    {
        PermissionFactory permissionFactory = (new DefaultPermissionFactory()).getFactory();

        permissionFactory.setProperty("RequiredPermission","com.ibm.resource.security.auth.ProfilePermission");
        permissionFactory.setProperty("RequiredPermission.name", "UserProfile");
        permissionFactory.setProperty("RequiredPermission.actions", "read");
        permissionFactory.setProperty("RequiredPermission.attributes", attributes);

        PrivilegedActionFactory privilegedActionFactory = (new DefaultPrivilegedActionFactory()).getFactory();
        privilegedActionFactory.setProperty("PermissionFactory", permissionFactory);
        privilegedActionFactory.setProperty("PrivilegedAction", "ReadAction");
        privilegedActionFactory.setProperty("attributes", attributes);

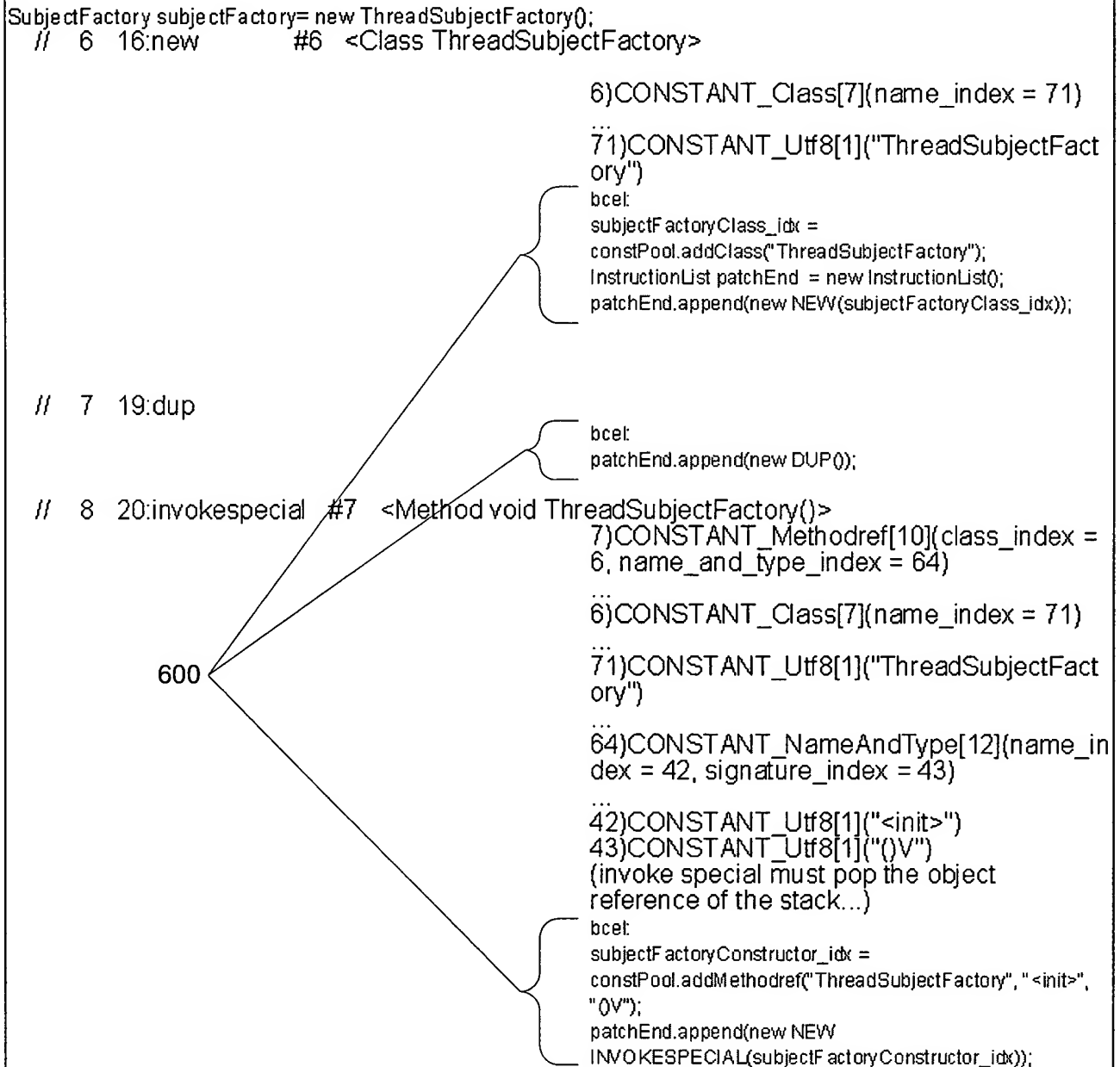
        Subject.doAsPrivileged(user, privilegedActionFactory.getPrivilegedAction(), null);
    }
    /*END INSTRUMENTATION */

    //get attributes associated w/profileName from DB
}
```

102b

Fig. 5

Byte-code instrumentation could be achieved using BCEL



102c

Fig. 6